

8月15日

```
private static final int QUICKSORT_THRESHOLD = 286;
private static final int INSERTION_SORT_THRESHOLD = 47;
private static final int COUNTING_SORT_THRESHOLD_FOR_BYTE = 29;
private static final int COUNTING_SORT_THRESHOLD_FOR_SHORT_OR_CHAR = 3200;
```

排序

java中Arrays.sort方法中的排序，为了达到企业级的方式法，能够用于生产。兼顾的多种排序算法的优缺点，编写了sort算法。在sort中，当数组长度低于286时，使用快排。快速排序进一步优化为两个privot的快速排序。一定程度上降低了代码递归的层数，增加代码的安全性，防止stackoverflow。在快排的区间元素小于46时，使用插入排序。插入排序为进一步优化的遍历一次插入两值的排序，减小了循环次数，提高性能。

pari排序的快速排序部分

1. 首先选取两个pivot，pivot的选取方式是将数组分成近视等长的六段，而这六段其实是被5个元素分开的，将这5个元素从小到大排序，取出第2个和第4个，分别作为pivot1和pivot2；
2. Pivot选取完之后，分别从左右两端向中间遍历，左边遍历停止的条件是遇到一个大于等于pivot1的值，并把那个位置标记为less；右边遍历的停止条件是遇到一个小于等于pivot2的值，并把那个位置标记为great
3. 然后从less位置向后遍历，遍历的位置用k表示，会遇到以下几种情况:
 - a. k位置的值比pivot1小，那就交换k位置和less位置的值，并是less的值加1；这样就使得less位置左边的值都小于pivot1，而less位置和k位置之间的值大于等于pivot1
 - b. k位置的值大于pivot2，那就从great位置向左遍历，遍历停止条件是遇到一个小于等于pivot2的值，假如这个值小于pivot1，就把这个值写到less位置，把less位置的值写道k位置，把k位置的值写道great位置，最后less++，great--；加入这个值大于等于pivot1，就交换k位置和great位置，之后great--
1. 完成上述过程之后，带排序的子区间就被分成了三段(pivot2)，最后分别对这三段采用递归就行了。

语法

新语法：直接打破a后面的代码块，即直接break两层循环。

```
a:
for () {
    for () {
        break a;
    }
}
```

跳表

跳表图解: <https://juejin.cn/post/6844903446475177998>

w: 写

e: 输入

r: 读

x: 输出